UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/618,810	07/14/2003	Ajay Kumar	5681-15200	5782

58467 7590 11/12/2008

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
WONG, JOSEPH D	
ART UNIT	PAPER NUMBER
2166	
MAIL DATE	DELIVERY MODE
11/12/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/618,810	KUMAR ET AL.	
	Examiner	Art Unit	
	JOSEPH D. WONG	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 March 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 July 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :20080326, 20070907, 20050211,20050701 , .

DETAILED ACTION

Drawings

Figures 1-10 are objected to because they are not formal with respect to 37 CFR 1.84 because the reference numerals are enclosed within an outline.

Specification

The abstract of the disclosure is objected to because it exceeds 150 words. Correction is required. See MPEP § 608.01(b).

Claim Objections

Claims 30 and 48 are objected under 37 CFR 1.83 for not appearing to illustrate the feature "resuming the transaction manager in response the resume request by granting read locks on the transaction freeze object". Consequently dependent claims 31-38, 48-56 are objected also.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re*

Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1, 10, 11, 20, 21, 30, 39 and 48 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-36 of copending Application No. 10/618,828. Although the conflicting claims are not identical, they are not patentably distinct from each other because it appears commonly owned with common inventors and the claimed subject matter is appears to differ superficially. In claim 1, the instant application adds, "wherein for each transaction". Other synonymous phraseology or obvious variations are present such as instant claim 1 recites "not change the state of the transaction without said permission" and the copending claim 1 recites "does not allow the one or more transactions to complete".

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Objections

Claims 21, 30, 39 and 48 are objected to for having minor punctuation informalities under MPEP 608.01(m).

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 21-56 are rejected for being directed towards nonstatutory subject matter.

Claim 21 is directed to a method of "receiving a pause request; pausing a transaction manager...receiving a plurality of resume requests; and resuming the transaction manager". The claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. Merely receiving a request does not necessarily convey a perceptible result to another. More specifically, the claimed subject matter provides for "pausing" and "resuming" without result after the intervening method is evidence that the result has been removed from the claim. A method claim without a useful, concrete, tangible result is nonstatutory under present evaluation. As such there this method does not produce a result thus, does not achieve the required status of having real world value. **Claims 22-19** depend from claim 21 and are rejected accordingly.

Claim 30 is directed to a method of receiving a pause request...pausing a transaction manager...receiving a resume request...resuming..." See above. More specifically, the claimed subject matter provides for "pausing" and "resuming" without result after the intervening method is clear evidence that the result has been removed from the claim. A method claim without a

useful, concrete, tangible result is nonstatutory under present evaluation. **Claims 31-38** depend from claim 30 and are rejected accordingly.

Claims 39-56 are directed to a carrier medium which are non-statutory under present evaluation because they appear directed toward signal bearing mediums such as transmission or communication media. The claim clearly recites "carrier medium". Instant specification paragraph [67] as published recites "transmission medium or signals such as electrical, electromagnetic, or digital signals, conveyed via communication medium such as network and/or wireless link". Such recitations are evidence that the medium includes signals and as such the claimed invention is drawn to a form of energy. Energy is not one of the four categories of invention. Energy is not a series of steps or acts and thus is not a process. Energy is not a physical article and as such is not a machine or manufacture. Energy is not a combination of substances and therefore is not a composition of matter. Furthermore the instructions are not necessarily executed upon a functional element of an operable computer such as a processor. Therefore claims 39-56 are not statutory under present evaluation.

Applicants can look to MPEP 2106.01-2106.02 (September 2007), Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility, Instant Specification, and contemporary case law with a matching fact pattern for further suggestions that may be helpful in overcoming these rejections.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1, 10-11 and 20 are rejected under 35 U.S.C. 102(b) as being anticipated by Hagersten et al, (US 5,983,326), hereinafter Hagersten.

As to claim 1, Hagersten teaches a system (Fig. 1), comprising: one or more processors (Col. 4, Lines 51-56, "multiprocessing computer system includes a plurality of processing nodes interconnected by an interconnect network"; Col. 7, Lines 14-16, " SMP node 12 includes multiple processors"); memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement: one or more applications configured to initiate one or more transactions (Col. 8, Lines 20-24, "memory operation...causing transfer of data from a source to a destination...within the initiator"; Col. 9, Lines 10-11, "Memory 22 is configured to store data and instruction code for use by processors"), wherein each of the one or more transactions comprises requests to access one or more data sources (Fig. 1, item 22, "Memory", nodes 12A-12D; Col. 9, Lines 6-9, "address and data phases of a transaction may be identified via a unique tag"); and a transaction manager configured to manage the one or more transactions initiated by the one or more applications (Col. 4, Lines 62-66, "home agent is configured to service multiple requests

simultaneously...transaction blocking unit is coupled to a home agent control unit for preventing the servicing of a pending coherent transaction request"); wherein for each transaction (Col. 1, Lines 19-22; Col. 8, Lines 26-27, read and write operations), the transaction manager is configured to request permission to change the state of the transaction (Fig. 7, "Receive Request...Check Block Status...Set Blocked Status...Write Reply...Write Data...Clear Block Status"), and wherein the transaction manager is configured to not change the state of the transaction without said permission (Fig. 11, "NACK...Not Acknowledge" or "NOPE...Negative Response").

As to claim 10, Hagersten teaches a system (Fig. 1), comprising a plurality of computer systems coupled by one or more networks (Fig 1B, item 38, "network"; Fig. 1, item 24, "System interface", item 20, "SMP BUS 20" meets network), wherein the plurality of computer systems comprise: one or more processors (Fig. 1, items 16A-16B, "P...P" for processors); and memory (Fig. 1, item 22, "memory") coupled to the one or more processors (Fig. 1, item 16A-16B, "P...P") and configured to store program instructions (Col. 3, Lines 10-15, 33-38, "if failed begin goto top end") executable by the one or more processors to implement one or more application servers comprising: one or more applications (Fig. 3, item 102, "home agent"; item 100, "request agent"; item 98, "transaction filter") configured to initiate one or more transactions (Col. 6, Lines 30-33, "operation performed in response to a read to own request from a processor"; Fig. 1-2), wherein each of the one or more transactions comprises requests to access one or more data sources (Fig. 1, item 22A-D, "memory" , nodes 12A-12D); and one or more transaction managers configured to manage the one or more transactions initiated by the one or more applications (Fig. 3, item 102, "home agent"...item 100, "request agent"); wherein for each

transaction (Fig. 4, item 110, "request"; Fig. 6, "request active"), the one or more transaction managers are configured to request permission to change the state of the transaction (Fig. 6, item 144, "Request Active"; item 150, "Write Active"), and wherein the one or more transaction managers are configured to not change the state of the transaction without said permission (Fig. 6, "Ignored Write").

As to claim 11, Hagersten teaches a system, comprising: one or more processors (Fig. 1); memory (Fig. 1, item 22, "memory") coupled to the one or more processors (Fig. 1, item 16A-B, "P....P") and configured to store program instructions executable by the one or more processors to implement: one or more applications (Fig. 3, "home agent") configured to initiate one or more transactions (Fig. 3, item 102, "home agent" or item 100, "request agent" or item 104, "slave agent"), wherein each of the one or more transactions comprises requests to access one or more data sources (Fig. 1, item 22A-D, "memory"); and a transaction manager configured to manage the one or more transactions initiated by the one or more applications (Fig. 3, item 102, "home agent"...item 100, "request agent"); wherein for each transaction ( ), the transaction manager is configured to request a read lock on a transaction freeze object to change the state of the transaction (Fig. 6, item 144, "Request Active"; item 150, "Write Active"), and wherein the transaction manager is configured to not change the state of the transaction without said lock (Fig. 6, "Ignored Write").

As to claim 20, Hagersten teaches a system (Fig. 1), comprising a plurality of computer systems coupled by one or more networks (Fig. 1A, item 38, "network"; Fig.1, item 20, "bus" or "system interface"), wherein the plurality of computer systems comprise: one or more processors (Fig. 1, items 16A-16B, "P....P", for processors); and memory (Fig. 1, item 22, "memory")

coupled (Fig. 1, item 20, "bus") to the one or more processors (Fig. 1, item 16A-B, "P....P") and configured to store program instructions executable (Fig. 9, 10, "read to share" or "read to own"; where read and own are instructions) by the one or more processors (Fig. 1, items 16A-B) to implement one or more application servers (Fig. 15A, "home agent control unit"; Fig. 15A, "directory cache management unit") comprising: one or more applications (Fig. 3, "home agent") configured to initiate one or more transactions (Fig. 9, 10, "read to share"), wherein each of the one or more transactions comprises requests to access one or more data sources (Fig. 1, items 22A-D, memories in nodes 12A-12D); and one or more transaction managers configured to manage the one or more transactions initiated by the one or more applications (Col. 8, Lines 20-24, "memory operation...causing transfer of data from a source to a destination...within the initiator"; Col. 9, Lines 10-11, "Memory 22 is configured to store data and instruction code for use by processors"; Fig. 15A, "directory cache management unit"); wherein for each transaction (Fig. 9, 10, "read", "share", "own"), the one or more transaction managers are configured to request a read lock on a transaction freeze object (Col. 4, Line 26, "spin-lock operations" or Line 33-36, "barrier synchronization...waiting CPUs") to change the state of the transaction (Fig. 14, 15, "transaction blocking unit" or Fig. 7, item 166, "IDLE" or Fig. 8, item 188), and wherein the one or more transaction managers are configured to not change the state of the transaction without said lock (Col. 4, Line 26, "spin-lock operations"; Col. 4, Lines 42-44, "delaying the access of the last CPU").

Claims 39-41, 43, 45, 47 are rejected under 35 U.S.C. 102(b) as being anticipated over Ault et al, (US 6,237,019), hereinafter Ault.

As to claim 39, Ault teaches a carrier medium comprising program instructions, wherein the program instructions are computer-executable to: receive a pause request (Fig. 3); pause a transaction manager in response the pause request (Col. 2, Lines 48-62, "suspends calling thread and places it on a lock wait queue...by creating a queues") by withholding permission to change the state of one or more transactions managed by the transaction manager; receive a resume request (Fig. 2, "ACCESS RESOURCE SEMOP(SEMID,+1); Fig. 3, "GRANT SEMAPHORE", item 316 is a read lock); and resume the transaction manager in response to the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager (Fig. 2, SEMOP(SEMID, -1)->"GRANT ACCESS", item 222).

As to claim 40, Ault teaches the carrier medium as recited, wherein a transaction freeze manager grants and withholds said permission (Fig. 3, Col. 2, Lines 48-62, "suspends calling thread and places it on a lock wait queue").

As to claim 41, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is a part of the transaction manager (Col. 2, Lines 48-62).

As to claim 43, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received (Col. 2, Lines 40-43, "in order for the kernel semaphore logic to atomically update the semaphore value and maintain the wait queue").

As to claim 45, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to grant a state transition permission request (conditional) if the transaction manager is not paused (Fig. 2-3, 5A-5E).

As to claim 47, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to not grant requests (conditional) if the transaction manager is paused (Col. 2, Lines 44-47, "...when the kernel owns this lock, other callers of semop(sic) will be suspended waiting for this lock..").

Claims 21-25, 27, 29 are rejected under 35 U.S.C. 102(e) as being anticipated by Oeltjen et al, (US 2004/0225972 A1), hereinafter Oeltjen.

As to claim 21, Oeltjen teaches a method, comprising: receiving a pause request ([38], "commands that permit stepping through and editing code such as pause"); pausing a transaction manager in response to the pause request by withholding permission ([9], "all this is done manually and is understandably prone to many mistakes, both human and resulting from the interactions of the many components of the design...more errors result from tools failing to interact properly...yet...errors require debugging", since a compile or compatibility error necessarily prevent the program from starting , such an error necessarily blocks a pause or restart) to change the state of one or more transactions managed by the transaction manager ([38], "permit stepping...step...pause...kill"), receiving a plurality of resume requests ([38], "commands such as...next, ..., continue"); and resuming the transaction manager in response the resume request by granting permission to change the state of the one or more transactions managed by the transaction manager ([38], "permit stepping through").

As to claim 22, Oeltjen teaches .the method as recited, wherein a transaction freeze manager grants and withholds said permission (Fig. 7, item 718, "result...fail", where a test failure ends execution in a manner that excludes pause and resume options; see Fig. 1, item 52).

As to claim 23, Oeltjen teaches the method as recited, wherein the transaction freeze manager is a part of the transaction manager (Fig. 4, items 440, 470, 474, [38]).

As to claim 24, Oeltjen teaches the method as recited, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager from an administrative entity ([38], "flow manager 460 also gives a flow developer 424 or other user full control over the location within the flow to manually set the current flow").

As to claim 25, Oeltjen teaches the method as recited, wherein the transaction freeze manager is (interpreted to be a capability) configured to queue received state transition permission requests and transaction manager pause requests in the order received ([38], See Fig. 7, where queuing is broadly interpreted to include any depth of buffering even 1 because all operations within the real world require time and are not instantaneously, see [35]).

As to claim 27, Oeltjen teaches the method as recited, wherein the transaction freeze manager is (intended use) configured to grant a state transition permission request (conditional) if the transaction manager is not paused (Fig. 7, [38], "specifically commands that permit stepping through and editing code such as step, next, run, continue, as well as pause").

As to claim 29, Oeltjen teaches the method as recited, wherein the transaction freeze manager is configured to not grant requests if the transaction manager is paused ([37], "without

this feature and given that flows may call hundreds of tools and take weeks to execute, any change or disability of the executing computers would require starting the process again").

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 2-9, 12-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hagersten in view of Fowler et al, (US 4,502,116), hereinafter Fowler.

As to claim 2, Hagersten teaches the system as recited, further comprising a transaction freeze manager (Fig.14, 15A, "transaction blocking unit", Col. 3, Lines 10-15, Lines 32-38, "spin lock").

However, Hagersten does not expressly teach configured to pause the transaction manager in response to a pause request by withholding said permission and resume the transaction manager in response a resume request by granting said permission.

However, Fowler teaches (intended use) configured to pause the transaction manager in response to a pause request by withholding said permission and resume the transaction manager in response a resume request by granting said permission (Col. 5, Lines 40-53,"pause/resume

synchronization of the multiprocessor system"; Col. 8, Lines 52-56, "control signal to...effect resumption of program execution").

Hagersten and Fowler are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Hagersten and Fowler because it provides for a stable testing and debugging environment for a multiprocessor system that aids in the debugging of the total system as discussed in Fowler, Col. 1, Lines 66-67; Col. 2, Lines 1-2.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Hagersten and Fowler because it provides for a stable testing and debugging environment for a multiprocessor system that aids in the debugging of the total system as suggested in Fowler, Col. 1, Lines 66-67; Col. 2, Lines 1-2.

As to claim 3, Hagersten teaches the system as recited, wherein the transaction freeze manager is a part of the transaction manager (Col. 3, Lines 5-15; Col. 30-50).

As to claim 4, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to receive requests to pause the transaction manager from an administrative entity (Fig. 9, 10"ADM...Administrative").

As to claim 5, Hagersten teaches the system as recited, wherein the transaction freeze manager is configured to queue received state transition permission requests and transaction manager pause requests in the order received (Fig. 15A, item 406, "RTS"; Col. 11, Lines 47-53, "FIFO").

As to claim 6, Hagersten teaches the system as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order (Col. 11,Lines 47-53, "FIFO; Col 26, Line 40).

As to claim 7, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to grant the state transition permission request (conditional option) if the transaction manager is not paused (Fig. 7, "not blocked"->"set blocked status").

As to claim 8, Hagersten teaches the system as recited, wherein the transaction freeze manager is configured to grant the pause request if the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request (Fig. 7, "not blocked"->"set blocked status").

As to claim 9, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to not grant requests (conditional) if the transaction manager is paused (Fig. 7, "not blocked"->"set blocked status").

As to claim 12, see claim 2 above.

As to claim 13, see claim 3 above.

As to claim 14, see claim 4 above.

As to claim 15, see claim 5 above.

As to claim 16, see claim 6 above.

As to claim 17, see claim 7 above.

As to claim 18, see claim 8 above.

As to claim 19, Hagersten teaches the system as recited, wherein the transaction freeze manager is (intended use) configured to not grant locks (conditional) if a write lock on the transaction freeze object is currently held by an administrative entity (Fig. 9-10).

Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Oeltjen in view of Hagersten.

As to claim 26, Oeltjen merely suggests but does not expressly teach the method as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order ([35]).

However, Hagersten teaches the method as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

Oeltjen and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Oeltjen and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Oeltjen and Hagersten because it provides for a

multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as suggested in Hagersten, Col. 4, Lines 45-49.

Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Oeltjen in view of Armangau et al, (US 6,659,992), hereinafter Armangau.

As to claim 28, Oeltjen does not expressly teach the method as recited, wherein the transaction freeze manager is (intended use) configured to grant a transaction manager pause request if the transaction manager is not paused and there are no outstanding state transition permission requests received prior to the pause request.

However, Armangau teaches the method as recited, wherein the transaction freeze manager is (intended use) configured to grant a transaction manager pause request (conditional) if the transaction manager is not paused and there are (interpreted to be a negative limitation) no outstanding state transition permission requests received prior to the pause request (Abstract, "permit the data to be removed...buffer becomes empty, the data mover retrieves the overflow").

Oeltjen and Armangau are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Oeltjen and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as discussed in Armangau, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Oeltjen and Armangau because it provides for removing data

from the primary buffer at a faster rate than can be written to tape as suggested in Armangau, Abstract.

Claims 35, 44 and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault in view of Oliver and in further view of Hagersten.

As to claim 35, Ault merely suggests service queried lock requests in FIFO order [35] but stops short of clearly teaching the limitation.

Oeltjen and Oliver do not expressly teach the method as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

However, Hagersten teaches the method as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

Ault in view of Oliver and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as suggested in Hagersten, Col. 4, Lines 45-49.

As to claim 44, Ault merely suggests but does not expressly teach the carrier medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order ([35]).

Ault and Oliver do not expressly teach the carrier medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

Hagersten as applied above teaches the carrier medium as recited, wherein the transaction freeze manager is configured to service queued state transition permission requests and transaction manager pause requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

As to claim 53, Ault merely suggests but does not expressly teach the carrier medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order ([35]).

Ault and Oliver do not expressly teach the carrier medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order.

However, Hagersten as applied above teaches the carrier medium as recited, wherein the transaction freeze manager is configured to service queued lock requests in FIFO order (Col. 11, Lines 47-52; Col. 26, Lines 40-41).

Claims 30-32, 34, 36, 48-50, 52, 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault et al, (US 6,237,019), hereinafter Ault in view of Oliver, (US 6,029,190), hereinafter Oliver.

As to claim 30, Ault teaches a method, comprising: receiving a pause request (Fig. 3, Col. 2, Lines 48-62, "suspends calling thread and places it on a lock wait queue...by creating a queues"); pausing a transaction manager in response to the pause request by withholding read locks (supra) on a transaction freeze object receiving a resume request (Fig. 2, "ACCESS RESOURCE SEMOP(SEMID,+1); Fig. 3, "GRANT SEMAPHORE", item 316 meets a read lock); and resuming the transaction manager in response the resume request by granting locks on the transaction freeze object (Fig. 2, SEMOP(SEMID, -1)->"GRANT ACCESS", item 222).

However, Ault suggests a read lock in (Fig. 2, "SEMOP, SEMID, -1)->GRANT ACCESS") but does not clearly teach a "READ LOCK" necessarily occurs when accessing the semaphore (as READ can be reasonably interpreted as a verb and an adjective when any consideration of the semaphore variable).

Oliver clearly teaches read lock and write lock management system based upon semaphore availability (see Title).

Ault and Oliver are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as discussed in Oliver, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as suggested in Oliver, Abstract.

As to claim 31, Ault teaches the method as recited, wherein a transaction freeze manager grants and withholds the read locks (Fig. 3, item 360, "WAIT QUEUE FOR INTERNAL LOCK").

However, Ault suggests read locks (Fig. 2) but does not clearly teach "READ LOCK".

Oliver as applied above teaches read lock (see title).

As to claim 32, Ault teaches the method as recited; wherein the transaction freeze manager is a part of the transaction manager (Fig 2, 5A-C).

As to claim 34, Ault teaches the method as recited, wherein the transaction freeze manager is configured to queue received lock requests in the order received (Fig. 3, "wait queue", 5A).

As to claim 36, Ault teaches the method as recited, wherein the transaction freeze manager is configured to grant a read lock if the transaction manager is not paused (Fig. 3, 5A).

However, Ault does not expressly teach "read lock".

Oliver as applied above teaches read lock (see title).

As to claim 48, Ault teaches a carrier medium comprising program instructions (Fig. 3, 5A-5E), wherein the program instructions are computer-executable to: receive a pause request (Fig. 2, "suspend"); pause a transaction manager in response to the pause request by withholding locks (Fig. 5a, "do forever" necessarily withholds locks) on a transaction freeze object receive a resume request (Fig. 2, "resume waiters"); and resume the transaction manager in response to the

resume request by granting locks on the transaction freeze object (Fig. 2, "semop(semid, -1)->grant access").

However, Ault suggests read locks (Fig. 2) does not clearly teach read locks.

Oliver teaches read locks (see title).

As to claim 49, Ault teaches the carrier medium as recited, wherein a transaction freeze manager grants and withholds the locks (Fig 2, 5A-C).

However, Ault does not clearly teach read locks.

Oliver as applied above teaches read locks (set title).

As to claim 50, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is a part of the transaction manager (Fig 2, 5A-C).

As to claim 52, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to queue received lock requests in the order received (Col. 2, Lines 40-43, "in order for the kernel semaphore logic to atomically update the semaphore value and maintain the wait queue").

As to claim 54, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused (Fig. 3, 5A).

However, Ault does not clearly teach read locks.

Oliver as applied above teaches read locks (set title).

Claims 33, 38, 42, 51 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault in view of Oliver and in further view of Hagersten.

As to claim 33, Ault teaches the method as recited, wherein the transaction freeze manager is configured to receive requests (intended use) for locks on the transaction (Fig. 2, 5A-5C).

However, Ault does not clearly teach write locks.

Oliver teaches write locks (see Title).

Ault and Oliver are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as discussed in Oliver, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as suggested in Oliver, Abstract.

Also Ault and Oliver do not teach pause the transaction manager from an administrative entity.

However, Hagersten teaches pause the transaction manager from an administrative entity ([ ]).

Ault in view of Oliver and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Hagersten because it provides for a multiprocessing system employing an enhanced blocking mechanism for read-to-share transaction as suggested in Hagersten, Col. 4, Lines 45-49.

As to claim 38, Ault does not expressly teach the method as recited, wherein the transaction freeze manager is configured to not grant locks if (optional conditional) a lock on the transaction freeze object is currently held by an administrative entity.

However, Ault merely suggests a write lock (Fig. 2) but does not expressly teach a write lock.

Oliver teaches a write lock (see Title).

Ault and Oliver are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault and Oliver because it provides for read/write lock permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as discussed in Oliver, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for read/write lock

permits a plurality of reader threads to access protected data simultaneously, while only allowing a single writer thread to access to a protected data location as suggested in Oliver, Abstract.

However, Ault and Oliver do not expressly teach wherein the transaction freeze manager is configured to not grant locks if a lock on the transaction freeze object is currently held by an administrative entity.

Hagersten teaches wherein the transaction freeze manager is configured to not grant locks if a lock on the transaction freeze object is currently held by an administrative entity (Fig. 9, 10"ADM...Administrative").

Ault in view of Oliver and Hagersten are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Hagersten because it provides for multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as discussed in Hagersten, Col. 4, Lines 45-49.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault and Oliver because it provides for multiprocessing system employing an enhanced blocking mechanism for read-to-share transactions as suggested in Hagersten, Col. 4, Lines 45-49.

As to claim 42, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to receive requests to pause the transaction manager (Fig. 2, 5A-5C).

However, Ault and Oliver do not expressly teach pause from an administrative entity.

However, Hasgersten teaches pause from an administrative entity (Fig. 9).

As to claim 51, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to receive requests for (intended use) locks on the transaction freeze object to pause the transaction manager (Fig. 2, 5A-5C).

Ault suggests but does not expressly teach write locks (Fig 5A-5C).

However, Oliver as applied above teaches write locks (see title).

Ault and Oliver do not clearly teach an administrative entity.

However, Hagersten as applied above teaches an administrative entity (Fig. 9-10, "ADM....administrative").

As to claim 56, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is configured to not grant locks if a lock on the transaction freeze object is currently held by an entity (Fig 2, 5A-5E); pause the transaction manager from an entity (Fig. 2, 3).

However, Ault does not teach write lock.

Oliver as applied above teaches write lock (see title).

However, Ault and Oliver do not expressly teach administrative entity.

Hagersten as applied above teaches administrative entity (Fig. 9, "ADM...administrative").

Claims 37, 46 and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ault in view of Oliver and in further view of Armangau, (US 6,549,941), hereinafter Armangau.

As to claim 37, Ault teach the method as recited, wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused (Fig. 2, 5A-5E).

Ault does not teach the write and read lock.

However, Oliver teaches the write lock and read lock (see title).

Ault and Oliver do not expressly teach write lock and read lock and there are no outstanding lock requests prior to the lock request.

However, Armangau teaches there are no outstanding read lock requests received prior to the write lock request, and there are no outstanding read locks (Abstract).

Ault in view of Oliver and Armangau are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as discussed in Armangau, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Hagersten because it provides for removing data from the primary buffer at a faster rate than can be written to tape as suggested in Armangau, Abstract.

As to claim 46, Ault teaches the carrier medium as recited, wherein the transaction freeze manager is (intended use) configured to grant a transaction manager pause request if (optional condition) the transaction manager is not paused; wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused (Fig. 2-3, 5A-5E).

Ault does not expressly teach write lock and read lock.

However, Oliver teaches a write lock and read lock (see title).

Ault and Oliver do not expressly teach wherein the transaction freeze manager is configured to grant a lock if the transaction manager is not paused; and there are no outstanding locks.

However, Armangau teaches there are no outstanding read lock requests received prior to the lock request, and there are no outstanding locks (Abstract).

Ault in view of Oliver and Armangau are analogous art pertinent to the problem to be solved. A skilled artisan would have been motivated to combine Ault in view of Oliver and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as discussed in Armangau, Abstract.

Therefore at the time of invention, it would have been obvious to a person having ordinary skill in the art to combine Ault in view of Oliver and Armangau because it provides for removing data from the primary buffer at a faster rate than can be written to tape as suggested in Armangau, Abstract.

As to claim 55, Ault does not expressly teach the carrier medium as recited, wherein the transaction freeze manager is configured to grant a write lock if the transaction manager is not paused (Fig. 2-3, 5A-5E)

Ault does not expressly teach write lock and read lock.

However, Oliver as applied above teaches write lock and read lock (see title).

Ault and Oliver do not expressly teach there are no outstanding read lock requests received prior to the lock request.

Armangau as applied above teaches there are no outstanding read lock requests received prior to the lock request and there are no outstanding read locks (see Abstract).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph D. Wong whose telephone number is (571) 270-1015. The examiner can normally be reached on Mondays through Fridays from 10 AM – 6PM.

Applicant initiated interviews may be formally requested in advance by faxing a completed PTO-413A form to the examiner's personal fax number at (571) 270-2015. Form PTO-413A is used by the examiner to prepare for any proposed interview. A detailed agenda listing should be attached including any proposed claim language and/or arguments that will be presented. This form is used to determine whether any proposed interview would advance prosecution and fit within a prescribed time limit.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain T. Alam can be reached on (571) 272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://portal.uspto.gov/external/portal/pair. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JOSEPH D. WONG /JDW/
Asst. Examiner, Art Unit 2166
10 November 2008

/Y. W./
Primary Examiner, Art Unit 2169

/Hosain T Alam/

Supervisory Patent Examiner, Art Unit 2166